PARALLEL AND DISTRIBUTED SIMULATION OF FRAME RELAY NETWORKS

Andrzej Sikora and Ewa Niewiadomska-Szynkiewicz

Institute of Control and Computation Engineering,
Warsaw University of Technology
ul. Nowowiejska 15/19, 00-665 Warsaw, Poland
Research and Academic Computer Network (NASK)
ul. Wąwozowa 18, 02-796 Warsaw, Poland
e-mail: asikora@elka.pw.edu.pl, ens@ia.pw.edu.pl

Abstract: This paper deals with the description of a software framework FR/ASimJava for large scale Frame Relay networks simulation and its application to the exampled networks simulation. FR/ASimJava is implemented based on ASimJava – a Java-based software package that can be used to develop parallel and distributed simulators of complex real-life systems. The focus is on scalability and efficiency of large scale networks simulation. We describe the functionality, design and performance of FR/ASimJava software. The case studies – simulations of six networks operating under Frame Relay are provided to illustrate the effectiveness of the presented software tool.

Keywords: computer simulation, communication protocols, distributed simulation, simulators, software performance, synchronization

1. INTRODUCTION

Computer simulation is widely recognized as an important tool for researches and engineers that allow to analyze the behavior or and performance of networks and verify new ideas (new protocols, mechanisms, queuing schemes, network services, attacks, etc.). Although analytical methods allow to solve many problems, the complexity and scale of modern networks limit the applicability of purely analytic approaches. During recent years it became apparent that in systems engineering traditional research was supplemented by computational research. Even in the cases when analytical methods can be applied, a computer simulation is often used to verify and validate the results of formal analysis.

A variety of software environments simulating packets transmission through a network are available today. There are a number of possible sets of criteria that could be used for network simulators comparison, e.g. model size, execution time, memory requirements, scalability, programming interface, etc. Different tools are optimized with respect to different purposes. The comparative study of some popular simulators are reported in many papers, e.g. the results of the performance study involving: JavaSim www.javasim.org, ns-2 www.isi.edu/nsnam/ns, SSFNet-Java and SSFNet-C++ www.ssfnet.org are described by Nicol (2003), the comparison of OPNET www.opnet.com/products/modeler, ns-2 and JavaSim is concluded in study by Małowidzki (2004).

We are involved in development of large heterogenous networks simulating in near real time. The goal is to develop the simulator that can execute in real time, i.e. to simulate N seconds of network operation requires no more than N seconds of computer time. We plan to use this simulator to understand the true effects of new protocols, QoS provisioning mechanisms and new services. The main difficulty in packet level simulation is the enormous computational power, i.e. speed and memory requirements needed to execute all events involved by packets transmission through the network. The simulation requires a certain amount of memory to save the states of all nodes of the networks, so memory requirements increase very fast when consider large scale networks. Thus, the amount of memory limits the number of nodes that can be taken into consideration. Similarly, the time of simulation usually increases proportionally to the

size of network and traffic that is simulated. Another problem is scalability, i.e. how a given simulator scales for large topologies and high speed links. Parallel and distributed simulation has already proved to be very useful when performing the analysis of different real complex systems (Nicol and Fujimoto, 1994; Zeigler, *et al*., 2000). It allows us to reduce the computation time of the simulation program, to execute large programs that cannot be put on a single processor and to better reflect the structure of a physical system. Parallel and distributed execution of computations can improve the scalability of the network simulator both in terms of network size and execution speed, enabling large scale networks and more network traffic to be simulated in real time.

Different approaches to simulation partitioning can be considered. Time parallel simulation as described by Lin and Lazowska (1991), Wu, *et al*. (2003) assumes fixed sized problems simulation by partitioning time axis into intervals and, assign each processor to simulate the network over its assigned time period. Another approach – space parallel simulation, utilized in parallel discrete event methods (Nicol and Fujimoto, 1994; Zeigler, *et al*., 2000), assumes partitioning a large scale network into subnetworks that are mapped to a different processors. Presented in this paper Frame Relay simulator FR/ASimJava utilizes this approach. Finally, we can find the simulators that combine both time and space distribution, such as Genesis approach described by Szymanski, *et al*., (2002).

Last years a new paradigm for constructing parallel and distributed simulations was developed. It is based on the idea of federating disparate simulators, utilizing runtime infrastructure to interconnect them. Such implementation results in a collection of simulators (simulation entities) designated as federates. All simulators involved in a federation are viewed as black boxes. The runtime infrastructure software (RTI) used for simulators interconnecting is typically designed for coarse granularity concurrency. The RTI implements relevant services required by the federated simulation environment. The most important services are: simulators synchronization, secure and efficient communication and scalable platform architecture. The main advantage of federated simulators is high possibility of simulation models reuse. The discussion of federated paradigm to large scale simulation one can find in Fujimoto, *et al*. (2003) and the description of HLA (High Level Architecture) software system (www.dmso.mil/public/transition/hla). The federated approach was utilized in the presented Frame Relay networks simulator.

2. FRAME RELAY NETWORK SIMULATION

The FR/ASimJava simulator provides a framework for Frame Relay large scale networks simulations performed on parallel computers or networks of computers. It is implemented based on ASimJava (*Asynchronous Simulation Java*), a Java-based software tool for large scale systems simulation. Although ASimJava was described by Niewiadomska-Szynkiewicz and Sikora (2004), we provide a brief summary here, to make the paper self-contained.

2.1 ASimJava framework for simulation

The ASimJava (Asynchronous Simulation Java) is a Java-based framework for large physical systems simulation. It can be used to develop general purpose simulators designed as federations of disparate simulators, utilizing runtime infrastructure (RTI) to interconnect them. The ASimJava general structure enables to do discrete-event simulations that can be described in terms of logical processes (LPs) that communicate with each other through message-passing. LPs simulate the real life physical processes FPs. The synchronous and asynchronous variants of simulation are provided. In the case of asynchronous realization four (conservative, optimistic and hybrid) protocols are provided: conservative protocol with null messages (CMB) (Misra, 1986; Nicol and Fujimoto, 1994; Zeigler, *et al*., 2000), window conservative protocol (WIN) (Nicol and Fujimoto, 1994), optimistic Time Warp (TW) (Nicol and Fujimoto, 1994; Zeigler, *et al*., 2000) and hybrid Moving Time Window protocol (MTW) (Sokol *et al*., 1998).

The simulator that is built upon ASimJava classes has hierarchical structure. The simulated system is partitioned into several subsystems (subtasks), with respect to their functionality and data requirements. Each subsystem is implemented as a logical process (LP). Each LP can be divided into smaller LPs. Hence, the logical processes are nested, see Fig. 1. Calculation processes that belong to the same level of hierarchy are synchronized. The approach to synchronization depends on the chosen variant of simulation: global clock in synchronous simulation and one of four mentioned protocols (CMB, WIN, TW or MTW) in asynchronous simulation.

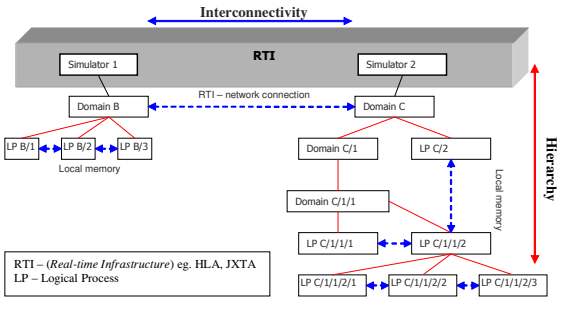

Fig. 1. A federation of network simulators developed upon ASimJava.

The configuration of the system to be simulated can be loaded and saved into the disc file in the XML format. It may contain any number of user-defined